International
Journal
Of Advanced Research in Engineering & Management (IJAREM)
ISSN: 2456-2033 || PP. 21-28

Applications of distribution function analysis of flow and connectivity in non-directed graphs for shortest path findings through the mathematical modelling of Floyd's algorithm

P.Kamal Devi[1], G.Eswara Prasad[2], V.S.Mathu Suresh[3]
[1]Assistant Professor, 2Head of the Department, 3Associate professor
Department of Mathematics

Abstract: Now a days graph theoretical models are most important in various applications in different fields. It is used to study the characteristics of molecules and atoms. The important advantages of graph theory is to calculate the actor's prestige in sociology. Routing is one vital application in computer science which is achieved through graph theory. It is one of the important problems in information forwarding from one point to another point in communication networks. So, it is evaluated through shortest path routing algorithm to transfer the information. A prominent amount of research has been evolved in seeking the solution for many problems in graph theory. Usually, some factors can be predetermined in the directed graphs such as vertices and edges. But, it is difficult to predetermine the vertices and edges in undirected graphs. The tentative variables such as connectivity and flow represents the correctness of distribution function. The number of complexity and searches of the procedure will be increased in the network, if number of nodes get increased. This paper aims to derive a mathematical modelling to find a shortest path distance based on Floyd's algorithm, which is applied for determining the flow, connectivity and distribution function in non-directed graphs. The different attributes in connectivity analysis can be applied for improving the real time network performance.

Keywords: Connectivity, Distribution function, Floyd's algorithm, Routing, Non-directed graphs.

1. Introduction

The advancements of computer age and the information system have stimulated more interest in the fundamental properties of the computer networks. The empirical study of the large networks had a profound impact due to large number of datasets can be easily stored and investigated. The analysis of complex networks plays a main role in science. The advanced research on the network topology has rapidly increased due to the success and evolution of internet [1].

The edges and vertices were well known to be predestined from earlier traditional diagram hypothesis [1] [2] and the parameters such as integration, diagram nature, and width determination depends on chart hypothesis were the major hypothetical issues. Hence to resolve the issues, in the recent decades an efficient assortment of proficient calculations have been proposed, which are used connected to a lot of people certifiable issues, for example, transportation, interchanges, and store network administration. Data on-existence remains a major problem of determination in current practice. The conventional methods have given a notion that it is not possible to estimate vertices and edges directly. In this paper, non-deterministic charts edges are considered, applied to depict the assembly of system also to manage it, likelihood hypothesis and arbitrary charts were developed [3] [4].

Usage 'n' is a private vertices and progressive edge among them (likelihood of $0<p<1$), hence the E&R arbitrary chart is developed. In this paper, initially the attributes of indeterminate chart width and then, obtain the comparing appropriation capacity [5] [6] was studied. Apart from this, it will execute a calculation from the Floyd's algorithm to estimate the circulation capacity, which will be demonstrated hypothetically and tentative. In the year 2007, uncertain theory was designed and improvised again by Liu [7] [8] is an excellent tool for non-deterministic resources (professional data). At present, the uncertain theory concept are used in network optimization and inventory problem transportation problem. In 2013, Gao and Gao introduced the concepts of uncertain theory into the graph theory [9]. Kruskal's [10] and Prim's [11] algorithm were derived with the intent to connectedness indexes. The objective of the current proposed paper is to focuses on the flow, connectivity and distribution function in non-directed graphs using enhanced algorithm based on Floyd's. In this paper, the characteristics of flow and connectivity in non-directed graph along with corresponding distribution function was proposed.

Major contribution of this paper is organized below. The section 1 covers an introduction about the graph theory. In Section 2, Floyd's algorithm is shortly introduced. Section 3 comprises the connectivity of indeterminate graphs with distribution function is obtained. The section 4 proposes a mathematical model with theorems for no-directed graph. An efficient framework for calculating the distribution function with illustrated numerical examples in section 5. The paper was concluded in section 6.

2. Floyd's algorithm

Floyd's algorithm is the extension of initial algorithm proposed by Bernard Roy [12] and are also known as Roy–Warshall algorithm, Roy–Floyd or 'WFI' algorithm which are used to identify the shortest paths using weighted graph during transition from +VE or –VE edge weights [13]. The proposed algorithm includes the comparison analysis of all selected paths from the vertices graphs. The all combination of edges were tested using $\psi\left(|Q|^2\right)$ edges in the graph. It incrementally improves the differences between the two vertices to identify the shortest path (until the estimate is optional).

Consider a graph 'G' with vertex (U). Furthermore, assume a shortest path (i,j,k) of function that returns the minimum conceivable path from i to j then use vertices starts from set {1, 2, 3….,k} (intermediate points) [9]. At present, the major objective is to identify the shortest path from each of i to j using only vertices 1; k+1.

Every path of vertices pairs have the possibility to achieve true shortest path; moreover through a path that only uses vertices in a set of {1, 2 …k} or the path is set from (i to k+1) & (k+1 to j).

Best path is derived from (i to j) that uses vertices through the 'k' is define by shortest path of i, j, k. the selected path is very clear. If a better path from I to k+1 to j. Then the shortest path is joining with length of (i to k+1 to j) and vertices in {1,…k} and from {k+1} to j.

In nominated vertices i & j, the desired shortest path is defined as (i, j, k+1) in a terms of recursive formula, and the formula is given below; when the weight is consider as 1(i, j); hence the base case is shortest Path (i, j, 0) = l (i, j).

Shortest Path (i, j, k+1) =minimum (shortest path (i,j,k),
Shortest path (i,k+1,k)+shortest path (k+1,j,k)

This formula is called as heart of Floyd's-warshall algorithm. The algorithm is processing with shortest path (i, j, k) for all (i, j) pairs for k=1 & k=2 and so on. This process continuous until k=N, and obtained the short path of (i, j) pair can use any transitional vertices [9] [10]. Basic version pseudo code is given as follows:

1) Let dist be an $|U|\times|U|$ array of minimum distances initialized to ∞ (infinity)
2) For each vertex U
3) Dist $[U][U]\leftarrow K$
4) For each edge (p, q)
5) Dist $[p][q]\leftarrow l(p,q)$ // the weight of the edge (p, q)
6) For k from 1 to $|U|$
7) For i from 1 to $|U|$
8) For j from 1 to $|U|$
9) If dist $[i][j]\leftarrow > dist[i][k]+dist[k][j]$
10) If Dist [i][j] $\leftarrow$ dist [i][j] + dist [k][j]
11) $dist[i][j]\leftarrow > dist[i][j]+dist[k][j]$

End

3. Non-Directed Graph and Its Connectivity

In this section includes the concepts of non-directed graphs with its connectivity and distribution function analysis of non-directed graphs.

3.1 Non directed graphs

The edges of non-directed graph or undirected graph possess no orientation. In a non-directed graph without loop, $n(n-1)/2$ provides the information about higher number of edges. These graphs possess each of three vertices and edges.

Basics of cycle matrix: The 'μ' fundamental cycles are formed by a chord with respect to some specified spanning tree, define a fundamental cycle matrix B f for a digraph. The orientation assigned to each of the fundamental cycles is chosen to coincide with that of the chord. Therefore B f, a i × j matrix can be expressed exactly in the same form as in the case of an undirected graph,

B f = [I i: B t]

Where, I i is the identity matrix of order i and the columns of B t correspond to the arcs in a spanning tree j.

3.2 Connectivity in non-directed graph

Every iterations of Floyd-Warshall algorithm recalculated in non-direct graph as in matrix form. Hence, it includes the lengths of paths and are gradually extending a set of all intermediate nodes. The first iteration of the process is employed to create matrix D_1, and step possess the paths among all the nodes using one (predefined) intermediate node. D_2 Contains lengths using 2 predetermined intermediate nodes. At last, the matrix D_n employs n intermediate nodes.

This transformation can be described as below:

$$D_{ij}^{n} = \min(D_{ij}^{n-1}, D_{ik}^{n-1} + D_{kj}^{n-1}) \qquad (1)$$

This transformation does not re-write the elements that are applied to estimate new matrix and can be used for same matrix for both D_i and D_{i+1}.

3.3 Distribution Function of Non-directed Graph

The distribution function of undirected graph G is a function gives the probability that can shows selective vertex in terms of degree k. $Q(k)$ can also be seen as the fraction of vertices in the graph that have degree k. similar definitions also apply for the in & out-links of vertices and it can be determine the degree distribution function for the outgoing links $Q_{out}(k)$ and an incoming one $Q_{in}(k)$.

For a random graph with connection probability p, the probability $Q(k)$ that a random node has degree k is given by,

$$Q(k) = \left(\frac{N-1}{k}\right) p^{k}(1-p)^{N-1-k} \qquad (2)$$

3.4 Objectives

The objectives are as follows,

- To derive a mathematical modelling to predict the shortest path from Floyd's algorithm
- To find the minimum paths in weighted graphs and estimate it in rapid manner.
- To develop an algorithm to calculate the shortest path.
- To reduce the computations in shortest paths

4. Mathematical Modelling And Theorems

Floyd's algorithm is working based on the following equations and theorems in non-directed graphs.

Theorem 1

Every graph is quasi-strongly connected, if it is connected with at least one root.

Theorem 2

Every graph is quasi-strongly connected, if it is has a spanning tree.

Theorem 3

The vertices of a graph will be sorted, if the graph is acyclic.

a) **Theorem 1**

Every graph is quasi-strongly connected, if it is connected with at least one root.

Proof

When it is root, digraph is connected as very strongly.

Assume, one quasi powerfully connected graph G and it must have one root. It is visible, if G is trivial. Or else, let assume the vertex set V= {v_1, v_2… v_n} of G where $n \geq 2$. The explanation of root is follows as,

1. Assume $R \leftarrow V$
2. Consider a path between two vertices such as u & v in R. If it is directed u-v –path means, then it can remove the v from R. The equation is given by,
$$R \leftarrow R - \{v\}$$
3. It will access the root, if there is only one vertex in R.
4. Consider a vertex s which is having two paths for vertices u and v are s-u path and s-v path. Now, s cannot be in R, if u is in R. now it can remove u and v in R and includes, i.e. $R \leftarrow R - \{u, v\}$ and $R \leftarrow R \cup \{s\}$.
5. Do the process many times as possible. From the final process, there are some vertices in R.

b) **Theorem 2**

Every graph is quasi-strongly connected, if it is has a spanning tree.

Proof

The root of T is considered as a root for G. It is quasi-strongly connected, if it has a spanning tree T. Consider, the G is connected with arc e. G is quasi-strongly connected, if e will be removed.

c) **Theorem 3**

The vertices of a graph will be sorted, if the graph is acyclic.

Proof

The vertices of graph will be sorted, only the graph is similar to acyclic. In cyclic graph, the vertices will not sorted by any values.

1. Let vertex (v) which is defined as a sink. Consider $\delta(v) \leftarrow n$, $G \leftarrow G - v$ and $n \leftarrow n - 1$.
2. Take $\delta(v) \leftarrow 1$, if only one vertex v in Graph G.

Let π be a path from the source vertex s to vertex v. Let $c(\pi)$ be the cost of path π using f. Similarly, let $d(\pi)$ remain the cost of π using g. Then $c(\pi) = \sum_{v_i \in \pi} f(v_i)$ and,

$$d(\pi) = \sum_{(v_i, v_i+1) \in \pi} g(v_i, v_{i+1}) = \sum_{(v_i, v_{i+1}) \in \pi} (f(v_i) + f(v_{i+1}))/2$$

$$\sum_{v_i} f(v_i) - (f(s) + f(v))/2 = c(\pi) - (f(s) + f(v))/2$$

Let π^* being minimum cost path from s to v utilizing f. Assume, without loss of generality, that $c(\pi^*)$ $< c(\pi)$. Then $c(\pi^*)$ - (f (s) + f (v))/2 < $c(\pi)$ - (f (s) + f (v))/2 and $d(\pi^*) < d(\pi)$.

5. Connectivity in Non-Directed Graphs - An Example

The following network weighted graph illustrates the path for each node (fig.1)

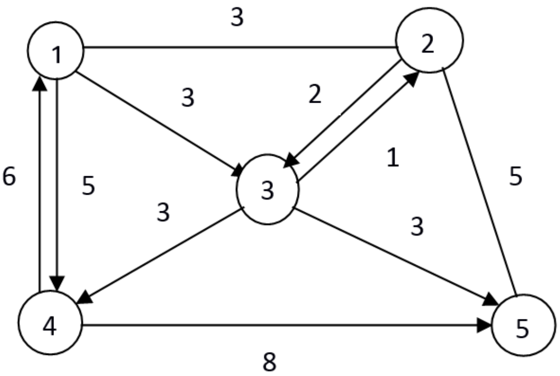

Fig.1. Network weighted graph for understanding Floyd's algorithim

The D_0 matrix formation of network weighted graphs is shown below in which the diagonal elements are 0 ($d^0_{ij} = 0$ for i = j) which indicates that there is no available path. The values indicate the each path weight ($d^0_{ij} \neq 0$ for $i \neq j$). The infinity indicates that there is no direct path.

$$D_0 = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{bmatrix} 0 & 3 & 3 & 5 & \infty \\ 3 & 0 & 2 & \infty & 5 \\ \infty & 1 & 0 & 3 & 3 \\ 6 & \infty & \infty & 0 & 8 \\ \infty & 5 & \infty & \infty & 0 \end{bmatrix}$$

The matrix Q_0 is obtained by identifying the shortest path between the processor node j, which leads to the next node i (for $i \neq j$).

$$Q_0 = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{bmatrix} - & 1 & 1 & 1 & 1 \\ 2 & - & 2 & 2 & 2 \\ 3 & 3 & - & 3 & 3 \\ 4 & 4 & 4 & - & 4 \\ 5 & 5 & 5 & 5 & - \end{bmatrix}$$

Where, i node is the intermediate predecessor of node j which leads to finding the short path from node i to j when $i \neq j$. For instance,

$$q^0_{2,1} = q^0_{2,3} = q^0_{2,4} = q^0_{2,5} = 2$$

By taking first logarithmic step, assuming k=1, the row elements of matrix D_1 can be determined.

$$d^1{}_{1,2} = \min\left|d^0{}_{1,2}; d^0{}_{1,1} + d^0{}_{1,2}\right| = \min[3; 0+3] = 3$$

$$d^1{}_{1,3} = \min\left|d^0{}_{1,3}; d^0{}_{1,1} + d^0{}_{1,3}\right| = \min[3; 0+3] = 3$$

$$d^1{}_{1,4} = \min\left|d^0{}_{1,4}; d^0{}_{1,1} + d^0{}_{1,4}\right| = \min[5; 0+5] = 5$$

$$d^1{}_{1,5} = \min\left|d^0{}_{1,5}; d^0{}_{1,1} + d^0{}_{1,5}\right| = \min[\infty; 0+\infty] = \infty$$

$$d^1{}_{2,1} = \min\left|d^0{}_{2,1}; d^0{}_{2,1} + d^0{}_{1,1}\right| = \min[3; 0+0] = 3$$

$$d^1{}_{2,3} = \min\left|d^0{}_{2,3}; d^0{}_{2,1} + d^0{}_{1,3}\right| = \min[2; 3+3] = 2$$

$$d^1{}_{2,4} = \min\left|d^0{}_{2,4}; d^0{}_{2,1} + d^0{}_{1,4}\right| = \min[\infty; 3+5] = 8$$

$$d^1{}_{2,5} = \min\left|d^0{}_{2,5}; d^0{}_{2,1} + d^0{}_{1,5}\right| = \min[5; 8+\infty] = 5$$

$$d^1{}_{3,1} = \min\left|d^0{}_{3,1}; d^0{}_{3,1} + d^0{}_{1,1}\right| = \min[\infty; \infty+\infty] = \infty$$

$$d^1{}_{3,2} = \min\left|d^0{}_{3,2}; d^0{}_{3,1} + d^0{}_{1,2}\right| = \min[1; \infty+8] = 1$$

$$d^1{}_{3,5} = \min\left|d^0{}_{3,5}; d^0{}_{3,1} + d^0{}_{1,5}\right| = \min[3; \infty+\infty] = 3$$

In matrix D_1, the matrix elements values are changed at the 4th column of 2nd row as 8, second column of 4th row as 9 and third column of 4th row as 9.

$$D_1 = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix}\begin{bmatrix} 0 & 3 & 3 & 5 & \infty \\ 3 & 0 & 2 & 8 & 5 \\ \infty & 1 & 0 & 3 & 3 \\ 6 & 9 & 9 & 0 & 8 \\ \infty & 5 & \infty & \infty & 0 \end{bmatrix}$$

Then, the following matrices were derived through the algorithm Q_1, Q_2, Q_2, D_2, Q_3, D_3, Q_4, D_5 and Q_5.

$$Q_1 = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix}\begin{bmatrix} - & 1 & 1 & 1 & 1 \\ 2 & - & 2 & 1 & 2 \\ 3 & 3 & - & 3 & 3 \\ 4 & 1 & 1 & - & 4 \\ 5 & 5 & 5 & 5 & - \end{bmatrix}$$

$$D_2 = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix}\begin{bmatrix} 0 & 3 & 3 & 5 & 8 \\ 3 & 0 & 2 & 8 & 5 \\ 9 & 1 & 0 & 3 & 3 \\ 6 & 9 & 9 & 0 & 8 \\ 8 & 5 & 7 & 13 & 0 \end{bmatrix}$$

$$Q_2 = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix}\begin{bmatrix} - & 1 & 1 & 1 & 2 \\ 2 & - & 2 & 1 & 2 \\ 2 & 3 & - & 3 & 3 \\ 4 & 1 & 1 & - & 4 \\ 2 & 5 & 2 & 1 & - \end{bmatrix}$$

$$D_3 = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix}\begin{bmatrix} 0 & 3 & 3 & 5 & 8 \\ 3 & 0 & 2 & 8 & 5 \\ 9 & 1 & 0 & 3 & 3 \\ 6 & 9 & 9 & 0 & 8 \\ 8 & 5 & 7 & 10 & 0 \end{bmatrix}$$

$$Q_3 = \begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix} - & 1 & 1 & 1 & 2 \\ 2 & - & 2 & 1 & 2 \\ 2 & 3 & - & 3 & 3 \\ 4 & 1 & 1 & - & 4 \\ 2 & 5 & 2 & 3 & - \end{bmatrix}$$

$$D_4 = \begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix} 0 & 3 & 3 & 5 & 8 \\ 3 & 0 & 2 & 8 & 5 \\ 9 & 1 & 0 & 3 & 3 \\ 6 & 9 & 9 & 0 & 8 \\ 8 & 5 & 7 & 10 & 0 \end{bmatrix}$$

$$Q_4 = \begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix} - & 1 & 1 & 1 & 2 \\ 2 & - & 2 & 1 & 2 \\ 2 & 3 & - & 3 & 3 \\ 4 & 1 & 1 & - & 4 \\ 2 & 5 & 2 & 3 & - \end{bmatrix}$$

$$D_5 = \begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix} 0 & 3 & 3 & 5 & 8 \\ 3 & 0 & 2 & 8 & 5 \\ 9 & 1 & 0 & 3 & 3 \\ 6 & 9 & 9 & 0 & 8 \\ 8 & 5 & 7 & 10 & 0 \end{bmatrix}$$

$$Q_5 = \begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix} - & 1 & 1 & 1 & 2 \\ 2 & - & 2 & 1 & 2 \\ 2 & 3 & - & 3 & 3 \\ 4 & 1 & 1 & - & 4 \\ 2 & 5 & 2 & 3 & - \end{bmatrix}$$

Matrices D_5 and Q_5 furnish us with completed information length of a shortest paths and the node on paths among all node pairs a transportation network. (e.g.) the shortest paths from, node 5 to 4 have length of 10. Also Q_5 matrix can obtain the shortest path from, from node 5 to 4. As the per Q_5, the immediate predecessor of node to node is 5 for the shortest path from 5 to 4, which is derived as $q_{54}^5 = 3$. Hence 3 to 4 forms the link on shortest path from 5 to 4. Following predecessor node to node 3 on the shortest path from 5 to 3 is $q_{53}^5 = 2$ .Hence 2 to 3 to 4 forms a part of the shortest path from 5 to 4. Preceding similarly, $q_{52}^5 = 5$. Hence 5 to 2 to 3 to 4 is the shortest path. Node 1 and 2 forms the link between 4 to 5.Hence 1 to 2 forms a part of the shortest path from 4 to 5. Hence 4 to 1 to 2 to 5 is also one shortest path.

6. Conclusion

In this paper, a mathematical modelling of Floyd's algorithm was derived which is applied for determining the flow, connectivity and distribution function in non-directed graphs. Initially, the theorems was developed for non-directed graphs which is inter related with Floyd's algorithm. The different attributes in connectivity are analyzed to determine the shortest path using weighted graph. There are three theorems, 1) Every graph is quasi-strongly connected, if it is connected with at least one root, 2) Every graph is quasi-strongly connected, if it is has a spanning tree, 3) The vertices of a graph will be sorted, if the graph is acyclic were constructed and proved in this paper which in uncorrelated with non-directed graph. The shortest path was also calculated from the derived mathematical formula based on Floyd's algorithm. It will widely use in computer science applications to develop the graphic algorithms. In future, this research extended even to improve the real time network performance.

References

[1]. A. Bargiela, W. Pedrycz, "Toward a theory of granular computing for human-centered information processing", IEEE Trans. Fuzzy Syst. 12 (2008) 320–330.
[2]. P. Bhattacharya, "Some remarks on fuzzy graphs", Pattern Recogn. Lett. 6 (1987) 297–302.
[3]. B. Bollobás, "Modern Graph Theory", Springer-Verlag, New York, 1998.
[4]. J. Chen, J. Li, "An application of rough sets to graph theory", Inf. Sci. 201 (2012) 114–127.
[5]. A. Dempster, "Upper and lower probabilities induced by a multivalued mapping", Ann. Math. Stat. 38 (2) (1967) 325–339.
[6]. S. Ding, "Uncertain multi-product newsboy problem with chance constraint", Appl. Math. Comput. 223 (2013) 139–146.
[7]. B. Liu, "Uncertainty Theory", second ed., Springer-Verlag, Berlin, 2007.
[8]. B. Liu, "Some research problems in uncertainty theory", J. Uncertain Syst. 3 (1) (2009) 3–10.
[9]. Floyd–Warshall algorithm. (2017, April 26). Retrieved May 01, 2017, from https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm.
[10]. X. Gao, Y. Gao, "Connectedness of uncertain graph", Int. J. Uncertainty Fuzz. Knowl.-Based Syst. 21 (1) (2013) 127–137.
[11]. Y. Gao, M. Wen, S. Ding, (s, S) "policy for uncertain single period inventory problem", Int. J. Uncertainty Fuzz. Knowl.-Based Syst. 21 (6) (2013) 945–953.
[12]. Y. Gao, "Uncertain models for single facility location problems on networks", Appl. Math. Model. 36 (6) (2012) 2592–2599.
[13]. Roy, Bernard (1959). "Transitivité et connexité." C. R. Acad. Sci. Paris. 249: 216–218.
[14]. Kenneth H. Rosen (2003). Discrete Mathematics and Its Applications, 5th Edition. Addison Wesley. ISBN 0-07-119881-4.
[15]. Kiruthika, R., & Umarani, R. (2012). Shortest path algorithms: A comparative analysis. International Journal of Management, IT and Engineering, 2(4), 55-62.